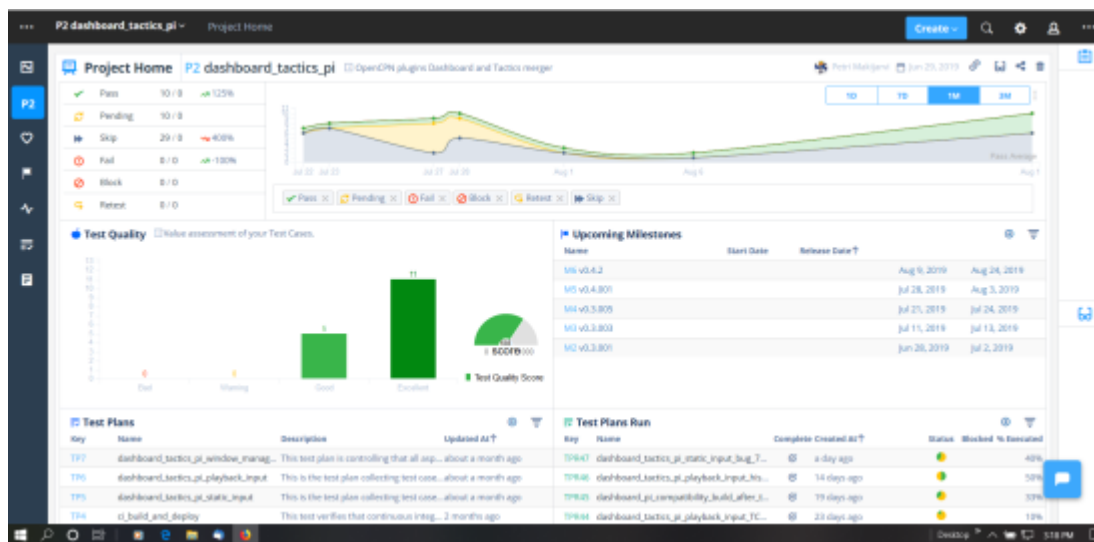


TestQuality -Test Management

Test Quality Website

Build test plans, test suites, test cases quicker
 Run test plans and capture test results more efficiently
 Automatically import test results from any test runner
 Gain critical insights quicker
 Communicate and coordinate with teams of any size
 FREE for open source projects on GitHub public repo's and ultra cost effective for private repo's.



How to use TestQuality

Get an account on testquality. From your github account online go to github marketplace or apps and search for Testquality [TestQuality Setup](#)

In your own testquality account, you can create projects as you like. Each project can be, for example, one of the plugins you are working on. You would link your GitHub repository to the testquality project. This is important.

First, you need to create a few major test plan categories (TP).

Then you would start to invent and plan use cases which can be made step by step, every time the same way with each step either PASS or FAIL, no ambiguity. These are called Test Cases (TC). Each Test Case must belong to a Test Suite (TS). Typically, you would execute all the test cases in a test suite, so don't make too big Test Suites. Each Test Suite must belong to a Test Plan.

Test Plan → (n) x Test Suites Test Suites → (n) x Test Cases

Don't worry about the steps in use cases too much. You can add new steps later on and then just make a

new test run:

You would create a Test Run, typically that contains all the tests in the Test Suite. Once you encounter a FAIL in a test step, stop: go to GitHub, create an issue for bug. It would have, say number #71. In your Test Case, you scroll down and you would Add a Defect. It asks you for a number, now give it number 71. You get the GitHub issue which you can resolve or leave for the future.

Test run = select Test -Suite → execute all (n) x Test Cases in it (Skipping allowed if you want to verify a fix of a single bug).

If you leave a bug fix for the future: after the fix, make a new Test Run, skip to the Use Case you are verifying the Defect # and then, at this failed step point, verify that it now PASS: In Defects click on the "71" defect: in the dialog mark it as resolved /fixed/closed and leave a comment. GitHub will be updated automatically.

It would help that if you define milestones, and use those for each test run, telling to which milestone (next milestone is usually next planned version) it belongs to. Then you would use in GitHub issues the same milestone number (in Issues).

Simple!

Collaborate on Test Quality

Good Testing is an intensive and complex process, and experienced help is always welcome. The more people testing the better the software. This tool promises to be very useful establishing and tracking the tests needed, defining, refining and implementing the necessary tests. It can communicate the tests needed to others and how they should be done step by step.

To collaborate and help on another person's TestQuality project, create a login on that person's testquality site [username].testquality.com using another or different email.

Here are some test Examples

- Example Test Process
- Test-current-spd.dir-w-leeway

Unfortunately this is missing some of the images which really help with understanding.

From:
<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:
https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:testquality

Last update: **2020/03/07 02:57**



